2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Establishing the NP-hardness of maximally permissive RAS-based approaches to multi-vehicle system safety

Elzbieta Roszkowska and Spyros Reveliotis

Abstract— **The safety problem in multi-vehicle systems seeks to establish collision-free and live vehicle motion, and it is a prominent problem for many configurations of these environments. Past work studying this problem in the context of free-range vehicular systems through abstractions based on Resource Allocation System (RAS) theory, has implicitly assumed that its resolution through maximally permissive supervision is NP-hard, and therefore, it has typically pursued suboptimal (i.e., more restrictive) solutions. The work presented in this paper offers formal proof to this implicit assumption, closing the apparent gap in the existing literature.**

I. Introduction

The safety problem in multi-vehicle systems seeks to establish collision-free and live vehicle motion, and it is a prominent problem for many configurations of these environments. In this work, we are particularly interested in the manifestation of this problem in the operational context of free-range vehicular systems, where a number of mobile agents travel concurrently along their respective paths within a confined planar area [1]. In the last years, this problem has received extensive attention with most of the proposed solutions adopting a continuous-time modeling approach; an indicative sample of this type of research are the works presented in [1], [2], [3], [4], [5], [6], [7], [8]. However, a significant limitation of (most of the) approaches based on continuous time modeling, is that they do not scale well to situations involving large fleets and/or complex operational environments. On the other hand, those approaches that resort to more distributed/decentralized computation in an effort to tame these complexity problems, frequently are of a more heuristic nature and they fail to provide the formal safety and/or liveness guarantees that might be necessary in many practical applications. Motivated by these remarks, the relevant research community has started considering solutions of a more "hybrid" nature, where the vehicle motion is largely planned and controlled in a more discretized space that results from an appropriate tessellation of the underlying physical terrain, while continuous-time motion models are used only for planning and coordinating the vehicle motion within each of the domains defined by the aforementioned tessellation; indicative examples of this type of work can be found in [9], [10].

In some of our recent work [11], [12], we have shown that, in the context of the aforementioned hybrid control paradigm, collision-free and live motion of the underlying traffic system can be established by superimposing a resource allocation structure on it and invoking results from the burgeoning resource allocation system (RAS) theory [13]. More specifically, under the approach(-es) considered in [11], [12], each agent is abstracted by a disk of radius ρ, and its overall motion profile is partitioned to a number of stages in a way that each of these stages implies the agent's exclusive access to a certain sub-space of the motion plane, big enough to cover the lane sector swept by the disk during the execution of this stage. This last requirement is further supported by the partitioning (or the "tessellation") of the motion plane into a number of "cells" that must be acquired and released by the agents sequentially and exclusively in order to execute their designated routes. While such a tessellation of the motion area can be performed in many ways, for reasons of convenience and tractability of the underlying dynamics, the aforementioned past works have considered rectangular tessellations; see also [14]. In this paper, similar to [14] and [12], we assume the partition of the plane into rectangular cells of side size at least 2ρ.[1] The cells of the resulting tessellation can be perceived as "resources" that must be acquired and released by the agents during the execution of their specified trips, and the aforementioned exclusivity of the cell allocation ensures the avoidance of the agent collision during their concurrent motion. At the same time, the enforcement of such an allocation paradigm arises the need for an additional control level – or for a "resource allocation protocol" – that will ensure that the applied resource allocation is "live", i.e., deadlocks will be avoided and every vehicle will eventually advance to its final destination.

As mentioned above, in the past literature, this requirement for live resource allocation in the considered traffic systems has been addressed through the adaptation of a set of results developed for the problem of deadlock avoidance arising in more generic resource allocation systems [13]. The derived resource allocation policies have been based on the implicit assumption that, similar to the more generic cases of sequential resource allocation, enforcing the liveness of the considered traffic systems in a maximally permissive manner is an NP-hard problem, and therefore, these policies tend to sacrifice permissiveness for computational tractability. However, when viewed from a more theoretical standpoint, the computational complexity of maximally permissive deadlock avoidance in the aforementioned traffic systems is an open issue. The topology of the agents' paths and the geometry of the tessellations employed in the specification of the resource allocation that takes place in these traffic environments, imply additional constraints for the structure of the resulting

E. Roszkowska is with the Institute of Computer Engineering, Control and Robotics, Wroclaw University of Technology, Poland, ekr@pwr.wroc.pl

S. Reveliotis is with the School of Industrial & Systems Engineering, Georgia Institute of Technology, USA, spyros@isye.gatech.edu

During the development of this work, Dr. Reveliotis was partially supported by NSF grants CMMI-0619978 and CMMI-0928231.

[1] The requirement that the size of the cell side is at least 2ρ, is introduced for the ease of exposition of the subsequent developments. Some further reflection on these developments will reveal that they hold true even when the aforementioned assumption is removed.

RAS. These constraints are not satisfied by the reductions that have been employed in the past for the establishment of the NP-hardness of maximally permissive deadlock avoidance / liveness-enforcing supervision in various RAS classes, and therefore, the relevant proofs are not directly applicable to the new cases considered herein.

This work addresses the theoretical gap identified in the previous paragraph. The presented results are structured as follows: In the next section we discuss the aforementioned tessellation of the motion plane, the induced partitioning of the agents' motion processes into (discrete) stages, and the resource allocation system that models the resulting discretized dynamics. Section III further formalizes these discrete dynamics through a DFSA model of the considered traffic system, and eventually, Section IV defines the (state) safety problem for free-range vehicles and proves its NP-completeness. Finally, we conclude by discussing the practical implications of this result and pointing out some other vehicle systems in which the complexity of the safety problem is still an open question.

II. The tessellation of the motion plane, the discretization of the agents' paths, and the induced RAS

We consider a set of autonomous mobile agents that move in a finite planar area $\mathcal{A} \subset \mathbf{R}^2$. Each agent is represented by a disk of radius ρ, and its center follows a pre-specified path that is given in the parametric form: $x^c = x^c(t)$, $y^c = y^c(t)$, $t \in [0, T]$. It is assumed that the agents stay off the system before they start their travel, and that they are retired from the system upon reaching their destination. However, during their concurrent motion in the system, the agents share the available space, and in order to avoid collisions, they may need to modify their velocity profiles. Such a coordination can be achieved through a hybrid control based on the *tessellation* of the motion plane into a number of areas, called *"cells"* [12]. Then, in the range of a cell, an agent controls its motion independently of the other agents, while cell crossing requires the permission of a supervisor, whose decisions depend on the system state, and which may temporarily prevent an agent from proceeding on its path.

More specifically, the motion area is abstracted as a grid of horizontal and vertical lines spaced at a distance $d \geq 2\rho$ and centered at the origin of the coordinate system (x, y). The resulting cells will be denoted by $W = \{w[i,j] : i \in \{-\underline{I}, \ldots, -\underline{1}, 0, 1, \ldots, \overline{I}\}, \ j \in \{-\underline{J}, \ldots, -1, 0, 1, \ldots, \overline{J}\}\}$, where $-\underline{I}$, $\overline{I}$, $-\underline{J}$, and $\overline{J}$ are taken large enough to encompass the entire (finite) area $\mathcal{A}$, that supports the agent motion. Then, given a point $(x, y) \in \mathcal{A}$ and a cell $w[i,j]$, we define

$$\begin{aligned}(x,y) \in w[i,j] \Longleftrightarrow & \ (i-1) \cdot d \leq x \leq i \cdot d \\ \wedge & \ \ (j-1) \cdot d \leq y \leq j \cdot d \end{aligned} \qquad (1)$$

The size d of the grid, that defines the length of the cell edges, should be selected by considering the efficiency of the system. In general, a smaller value of d can accommodate a larger number of agents, and therefore, can lead to a higher space utilization, but at the same time, it will lead to more disruption of the agent travels by the superimposed supervisory control, and possibly to more congested traffic and longer delays.

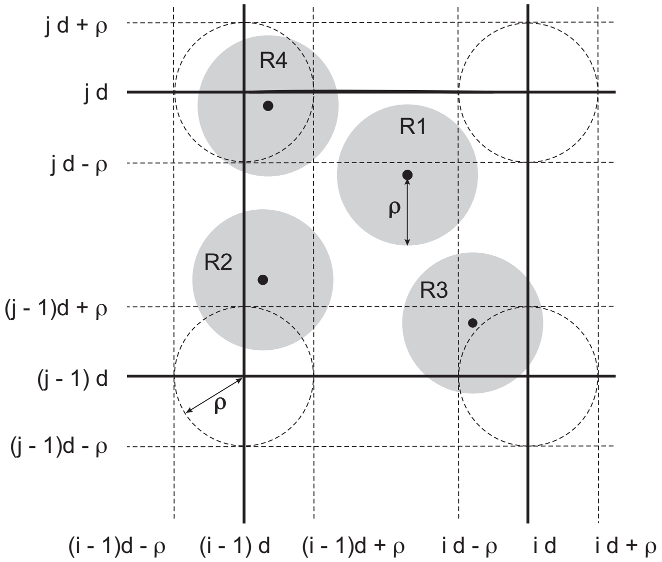

Fig. 1. The mapping $\mathcal{W}(C)$ and the partitioning of the motion place induced by it.

We shall say that an agent (with its disk) centered at (x^c, y^c) *lies in* cell $w[i,j]$ if and only if (*iff*) $(x^c, y^c) \in w[i,j]$.[2] On the other hand, we shall say that an agent centered at (x^c, y^c) *occupies* cell $w[i,j]$ *iff* there exists $(x, y) \in w[i,j]$ with $||(x,y) - (x^c, y^c)|| \leq \rho$, where $|| \cdot ||$ denotes the Euclidean norm.[3] Clearly, this definition induces a mapping $\mathcal{W}$ from the motion area, $\mathcal{A}$, to the powerset of W, 2^W, that maps to any point $(x, y) \in \mathcal{A}$ the cell subset $\mathcal{W}(x, y) \in 2^W$ consisting of the cells occupied by an agent centered at (x, y). A graphical illustration of this mapping $\mathcal{W}$ is given in Figure 1. We note that the adopted tessellation is defined by the grid of the solid horizontal and vertical lines, and the mobile agents are depicted by the grey disks in it. The reader should notice that an agent can occupy one cell (as in the case of $R1$), two neighboring cells (as in the case of $R2$), three neighboring cells (as in the case of $R3$), or four neighboring cells (as in the case of $R4$). We also notice that for the tessellation schemes considered in this work, the number of cells occupied by a mobile agent that is located at (x^c, y^c) is effectively determined by the relative positioning of (x^c, y^c) with respect to another partitioning of the motion plane, that is induced by the original tessellation scheme and the agent geometry. In Figure 1, this induced partitioning is defined by the depicted dashed lines and its detailed derivation, including a complete analytical characterization, can be found in [12].

In order to avoid collisions among the agents, it is required that at any point in time, a cell can be occupied by only one

[2] It should be noticed that according to Equation 1, an agent can lie in more than one cells at the same time. Especially, in the (rather singular) case that the agent center is located at the intersecting point of two grid lines, the agent will lie in all four neighboring cells.

[3] In order to maintain a simple notation, in the entire discussion of this manuscript we have assumed that the system agents are homogeneous with respect to their disk size. If, however, this is not the case, but each agent R_k occupies a disk of distinct radius, ρ_k, the concepts and structures defined in the rest of this section still apply, but they are customized for each agent through their parameterization by the agent radius ρ_k. Furthermore, all the results of the paper remain true provided that the grid size of the applied tessellation satisfies $d \geq 2 \max_k \rho_k$.

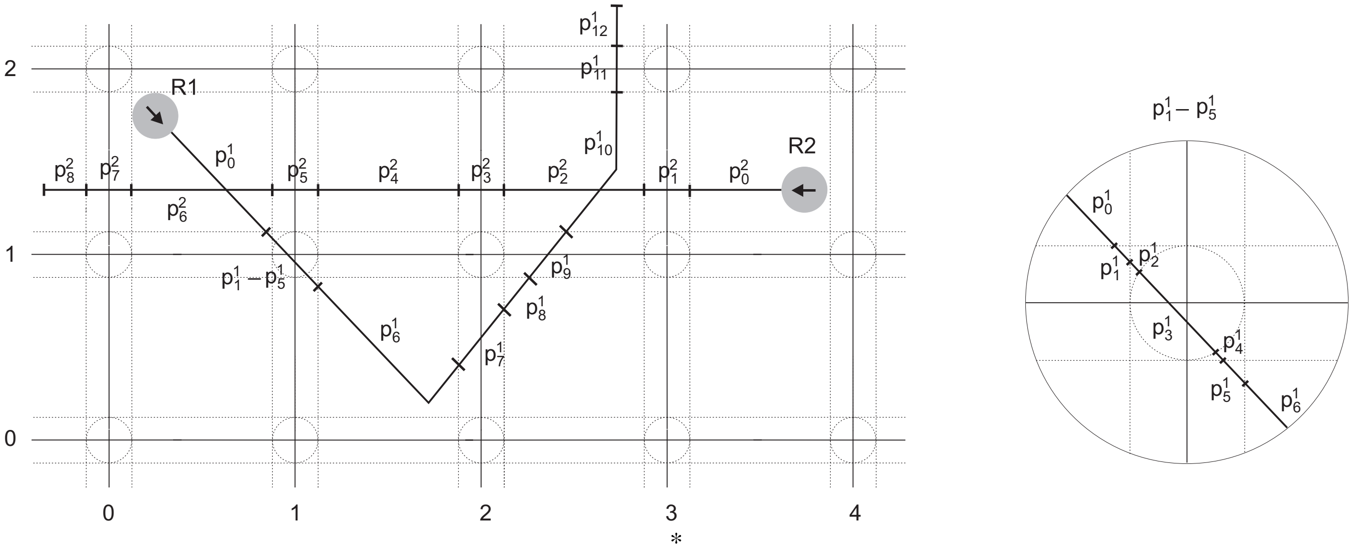

Fig. 2. Example paths for two mobile agents, and the corresponding resource allocation profiles that are defined by the path partitioning into maximal segments with the same cell occupation. The right part of the figure details the profile obtained for agent $R1$.

TABLE I
THE RESOURCE ALLOCATION INDUCED BY THE PATH SEGMENTATION OF FIGURE 2

	Stage No. j	Required resources $c(1, j)$
Agent R_1	0	$w[0, 1]$
	1	$w[0, 0], w[0, 1]$
	2	$w[0, 0], w[0, 1], w[1, 1]$
	3	$w[0, 0], w[0, 1], w[1, 0], w[1, 1]$
	4	$w[0, 0], w[1, 0], w[1, 1]$
	5	$w[0, 0], w[1, 0]$
	6	$w[1, 0]$
	7	$w[1, 0], w[2, 0]$
	8	$w[2, 0]$
	9	$w[2, 0], w[2, 1]$
	10	$w[2, 1]$
	11	$w[2, 1], w[2, 2]$
	12	$w[2, 2]$

	Stage No. j	Required resources $c(2, j)$
Agent R_2	0	$w[3, 1]$
	1	$w[3, 1], w[2, 1]$
	2	$w[2, 1]$
	3	$w[2, 1], w[1, 1]$
	4	$w[1, 1]$
	5	$w[1, 1], w[0, 1]$
	6	$w[0, 1]$
	7	$w[0, 1], w[-1, 1]$
	8	$w[-1, 1]$

agent. Hence, the cells defined by the proposed tessellation constitute *fictitious resources of unit capacity*. Furthermore, under the proposed zoning scheme, the paths designated to the different agents are naturally segmented to a number of *stages*, with each stage corresponding to a maximal path segment with constant cell (i.e., resource) occupation. The resulting stage sequences define the corresponding *resource allocation processes* that must be observed by each agent. In particular, in the proposed regime, an agent must secure the cells associated with a certain stage before it can proceed to the execution of the path segment corresponding to that stage. Also, in certain cases, an agent can enter a new stage of its path by simply releasing some of the cells held in its previous stage. Figure 2 exemplifies the abstracting notion of the resource allocation profile, by applying it on the motion profiles, p^1 and p^2, of two agents, R_1 and R_2. Path p^1 consists of thirteen (maximal) segments p^1_0 - p^1_{12}, and path p^2 consists of nine such segments, p^2_0 - p^2_8. Also, Table I specifies the cells occupied by the two agents at the various stages of their route.

It is clear from the above discussion that, in the proposed regime, the agent motion dynamics must be complemented with a resource allocation protocol that will enable each agent to acquire mutually exclusive access to the cells required at each particular stage of its motion process. Consequently, the considered system of free ranging agents is naturally abstracted to a sequential resource allocation system (RAS) according to the modeling paradigm of [13]. In particular, following the classification of [13], the resulting RAS presents strong similarity to the class of Linear-Conjunctive-RAS (L-CON-RAS), as it involves linear resource allocation sequences where, however, some processing stages might require the simultaneous – i.e., conjunctive – allocation of more than one resource units. On the other hand, a key attribute that differentiates the resource allocation taking place in the considered vehicle systems from the broader resource allocation schemes belonging into the L-CON-RAS class, stems from the fact that the resource allocation and/or de-allocation that takes place during the transition between two consecutive processing stages, must observe a "resource proximity" relation that is defined by the adopted tessellation. More specifically, in the considered RAS systems, the allocation corresponding to a particular processing stage must be interpretable as the occupation of a number of neighboring cells by the corresponding mobile agent, while the variation of the allocations between two consecutive processing stages must be interpretable as the occupation of some new neighboring cells and/or the release of some previously held ones, during the agent motion. The sub-class of L-CON-RAS that possesses the aforementioned additional features will be characterized as FREE-RANGE-RAS.

Formally, a FREE-RANGE-RAS can be specified by a triplet $\Phi = (W, P, D)$, where: (i) W is the set of the system resources (cells), (ii) $P = \{P_1, P_2, \ldots, P_n\}$ is the set of the system processes (agents' motion processes along their respective paths), where each process P_i, $i = 1, \ldots, n$, consists of $\Xi_{i1}, \Xi_{i2}, \ldots, \Xi_{il_i}$ consecutive *processing stages*, and (iii) $D : \Xi = \{\Xi_{ij} \mid i = 1, \ldots, n; j = 1, \ldots, l_i\} \rightarrow 2^W$ is the *resource allocation function* associating every processing stage Ξ_{ij} with the subset of resources required for its execution in the way satisfying the aforementioned constraints with respect to the underlying tessellation. It is further assumed that a process executing a non-terminal stage $\Xi_{ij}, i = 1, \ldots, n; j = 1, \ldots, l_i - 1$, must first be allocated the resources $D(\Xi_{i,j+1})$ in order to advance to its next stage $\Xi_{i,j+1}$, and only then it can release the no-more required resources $D(\Xi_{i,j+1}) \setminus D(\Xi_{ij})$. Finally, as stated earlier, the considered resource allocation protocol further requires that no resource is allocated to more than one process at a time.

The next section provides a formal characterization of the behavioral dynamics of FREE-RANGE-RAS by means of a deterministic finite state automaton (DFSA) [15]. However, before closing the discussion of this section, and for reasons that will become clear in the sequel, we distinguish a particular type of the agent paths that consist only of horizontal and vertical segments joining the centers of the consecutive cells that they lie on. We shall refer to these paths as *central vertical-horizontal paths*, and specify them by the sequence of the traversed cells, $p = w_1, w_2, \ldots, w_u$. The motion process of an agent that follows such a path consists of $2u-1$ stages that, respectively, require the following resource sets: $\{w_1\}$, $\{w_1, w_2\}$, $\{w_2\}$, $\ldots$, $\{w_{w-1}\}$, $\{w_{u-1}, w_u\}$, $\{w_u\}$. For example, the path depicted in Figure 3 is specified by the resource sequence $p = w[0,0], w[0,1], w[1,1], w[2,1], w[2,0]$ and consists of nine stages, which require the respective cell subsets: $\{w[0,0]\}$, $\{w[0,0], w[0,1]\}$, $\{w[0,1]\}$, $\{w[0,1], w[1,1]\}$, $\{w[1,1]\}$, $\{w[1,1], w[2,1]\}$, $\{w[2,1]\}$, $\{w[2,1], w[2,0]\}$ and $\{w[2,0]\}$. The reader should particularly notice that the specification of a central vertical-horizontal path $p = w_1, w_2, \ldots, w_u$ determines uniquely the underlying resource allocation process.

III. A DFSA-based representation of the considered traffic systems

This section provides a formal characterization of the qualitative dynamics of the considered traffic systems and their behavioral properties of interest in this work, by employing a variation of the well known *deterministic finite-state automaton* (DFSA) [15]. Hence, in the sequel, first we introduce the considered DFSA model, and subsequently we employ this modelling framework in order to characterize the feasible and the desirable behavior of the traffic systems under consideration.

Automata – otherwise known as *state machines* – provide a convenient, general tool for abstracting the qualitative behavior of discrete event systems (DES). In this work, we shall employ a particular sub-class of this model, that is formally defined as follows:

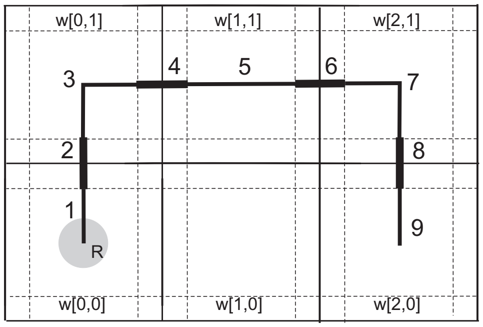

Fig. 3. An example of a central vertical-horizontal path. The path is uniquely specified by the sequence of the traversed cells, $p = w[0,0], w[0,1], w[1,1], w[2,1], w[2,0]$, and it induces nine stages for the agent's motion process.

Definition 1: A *deterministic finite state automaton (DFSA)* is defined as a tuple $G = (S, E, \delta, s_0, S_M)$ such that:

1) S and E are *finite* sets, respectively known as the *state* and the *event* set of the automaton.
2) $\delta : S \times E \rightarrow S$ is a *partial* function, known as the *state transition function* of the automaton.
3) $s_0 \in S$ and $S_M \subseteq S$ are, respectively, the *initial* state and the set of marked states of the automaton.[4]

The above DFSA starts its operation from state s_0. In each state $s \in S$, an event e can only occur if the state transition function $\delta()$ is defined on the pair (s, e); in that case, we say that event e is *enabled* at s. The occurrence of event e at s results in a new state $s' = \delta(s, e)$, which can be changed subsequently by the occurrence of event e' that is enabled in state s', and so on. In order to capture state transitions arising from strings of events, the state transition function δ can be naturally extended to $S \times E^*$ as follows:

$$
\begin{aligned}
\forall s \in S, \quad & \delta(s, \epsilon) && \equiv s \\
\forall s \in S, \forall u \in E^*, \forall e \in E, \quad & \delta(s, ue) && \equiv \delta(\delta(s, u), e)
\end{aligned}
$$

In the above equation, ϵ denotes the empty string, and E^* denotes the set of all strings that can be constructed with the elements of the set $E \cup \{\epsilon\}$. Moreover, it is implicitly assumed that the involved single-step transitions correspond to enabled events, i.e., to state-event pairs for which the original function δ is defined; otherwise, the extended version of δ is undefined on the corresponding state-string pair. We say that state $s' \in S$ is *reachable* from state $s \in S$ if there exists string $u \in E^*$ such that $s' = \delta(s, u)$; the set of all states reachable from s is called the *reachability set* of s and denoted by $R(s)$. The particular set $R(s_0)$ is also recognized as the reachability set of the DFSA G. Graphically, the dynamics of G can be represented by a directed multi-graph $RG(G) = (V, F)$, called the *reachability graph* of G; the vertex set V of this graph is defined by $R(s_0)$ and its edge set F is the subset of $R(s_0) \times E \times R(s_0)$ such that edge $f = (s, e, s') \in F \iff s' = \delta(s, e)$. Event e is typically perceived as the "label" of edge (s, e, s').

[4] Typically, from a modeling standpoint, arrival to a marked state denotes the completion of a certain task.

The DFSA $G(\Phi) = (S, E, \delta, s_0, S_M)$ abstracting the feasible dynamics of a FREE-RANGE-RAS $\Phi = (W, P, D)$ is defined as follows:

1) The state set S consists of all vectors $s = [s_{11}, s_{12}, \ldots, s_{1,n_1}, s_{21}, \ldots, s_{2,n_2}, \ldots, s_{n1}, \ldots, s_{n,n_n}] \in \{0,1\}^{|\Xi|}$ such that:
   - for each $i = 1, \ldots, n$, $\sum_{j=1}^{n_i} s_{ij} \in \{0, 1\}$.
   - for each $i = 1, \ldots, n$, each $q = 1, \ldots, n$, each $j = 1, \ldots, n_i$, and each $r = 1, \ldots, n_q$, it is true that if $s_{ij} = s_{qr} = 1$ and $(i, j) \neq (q, r)$ then $D(\Xi_{ij}) \cap D(\Xi_{qr}) = \emptyset$.

   Each entry $s_{ij} = 1$ indicates that agent R_i is on the j-th stage of its motion process, P_i.
2) The event set is given by $E = \{e_{ij} \mid i = 1, \ldots, n; j = 0, \ldots, n_i\}$, where event e_{i0} represents the *start* of process P_i, event e_{in_i} represents the *end* of process P_i, and event e_{ij}, $j \in 1, \ldots, n_i - 1$, represents the *advancement* of P_i from stage $\Xi_{i,j}$ to stage $\Xi_{i,j+1}$.
3) For each pair (s, e_{qr}) s.t. the state transition function δ is defined, the value of the components s'_{ij} of the new state $s' = \delta(s, e_{qr})$ is given by:
$$s'_{ij} = \begin{cases} s_{ij} - 1 & \text{if} \quad i = q \quad \text{and } j = r \\ s_{ij} + 1 & \text{if} \quad i = q \quad \text{and } j = r + 1 \\ s_{ij} & \text{otherwise} \end{cases}$$
4) $\delta(s, e_{qr})$ is defined if the tentative state $s' \in S$.
5) The *initial state* $s_0 = \mathbf{0}$, which corresponds to the situation when the system is empty of any processes.
6) The set of marked states is the singleton $S_M = \{s_0\}$.

The designation of state s_0 as the only marked state of the considered DFSA expresses the requirement that all activated vehicles must proceed to the completion of their trip and retire from the system. States that provide this capability are characterized as safe. In the next section we provide a formal definition of the notion of (state) safety and investigate its computational complexity.

IV. The state safety problem and its complexity

The state safety problem for the vehicular system introduced in the previous sections can be formally stated as follows.

FREE-RANGE-RAS state safety: Given a FREE-RANGE-RAS specified by the triplet $\Phi = (W, P, D)$, the induced DFSA $G(\Phi) = (S, E, \delta, s_0, S_M)$ and a state $s \in S$, is the initial state s_0 reachable from s?

The main contribution of this work is to establish that in the considered operational regime, the above problem is NP-complete [15]. This is stated and proven in the sequel.

Theorem 1: For rectangular tessellations with step sizes greater than or equal to 2ρ, the problem of FREE-RANGE-RAS state safety is NP-complete in the strong sense.

Proof. In order to prove the theorem,

1) first we show that the considered FREE-RANGE-RAS state safety problem belongs to the problem class $\mathcal{NP}$,
2) and subsequently we establish its NP-completeness by reducing to it the well-known NP-complete problem of 3-SAT [15].

Proof of (1): We remind the reader that a decision problem is in the class $\mathcal{NP}$ *iff* it can be solved in polynomial time by a Nondeterministic Turing Machine (NDTM) [15]. Notice that the size of an instance of the considered problem is essentially determined by the size of the data required to specify function D, which is proportional to the total number of processing stages, $|\Xi|$. Since this number bounds also the length of any event sequence σ that can constitute a feasible solution to the considered state safety problem, it follows that the problem can be solved by an NDTM in polynomial time, and consequently the FREE-RANGE-RAS state safety problem belongs to the class $\mathcal{NP}$.

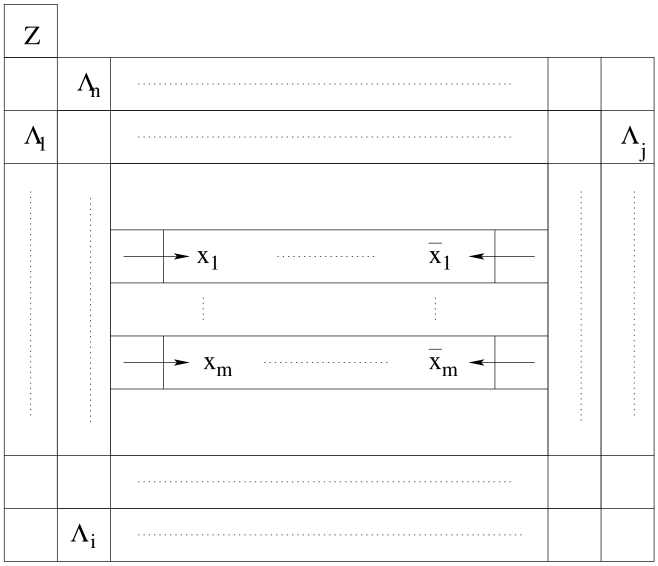

Fig. 4. Illustration for the proof of Theorem 1: Highlighting the basic topology of the agent paths that define the FREE-RANGE-RAS employed in the relevant reduction from 3-SAT.

Proof of (2): As mentioned above, in order to prove the NP-completeness of the FREE-RANGE-RAS state safety problem considered in this theorem, we will provide a reduction from the 3-SAT problem. We remind the reader that the 3-SAT problem can be stated as follows:

3-SAT [15]: Given a set of literals $\mathcal{X} = \{X_1, \bar{X}_1, X_2, \bar{X}_2, \ldots, X_\mu, \bar{X}_\mu\}$ and a set of clauses $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_\nu\}$, each clause being a disjunction $\Lambda_q = y_q^1 \vee y_q^2 \vee y_q^3$, $y_q^1, y_q^2, y_q^3 \in \mathcal{X}$, does there exist $K \subseteq \mathcal{X}$ such that the conjunction of the clauses in Λ is satisfiable, i.e., 1) $\forall i = 1, \ldots, \mu$, K does not contain both X_i and $\bar{X}_i$, and 2) $\forall q = 1, \ldots, \nu$, $K \cap \Lambda_q \neq \emptyset$?

The proposed reduction will construct from any 3-SAT problem istance $(\mathcal{X}, \Lambda)$, an instance of the considered FREE-RANGE-RAS state safety problem, with $\Phi = (W, P, D)$ and state $s \in S$, as follows:

a) The set of resources W consists of the set of (square) cells depicted in Figure 4.
b) The set of processes is given by $P = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_\nu\} \cup \{Z\}$.
c) The resource allocation function D is implied by the topology of the paths followed by the agents, which involves (c.f. Figure 4) :
   - two nested rings,
   - μ "bridges", each corresponding to one of the variables $X_1, \ldots, X_\mu$ of the 3-SAT problem and consisting of ν cells,
   - another cell, marked by Z in the figure.

   The agents executing the processes Λ_q, $q = 1, \ldots, \nu$, are distributed at various (arbitrary) locations at the outer ring and each of them has to pursue the following central vertical-horizontal path:

i) First it enters the inner ring through the cell of that ring which is next to it.

ii) Subsequently it moves clockwise on that ring until it meets the entry point of the bridge corresponding to the first variable in clause Λ_q. If this variable is not negated in the clause, the bridge must be crossed from left to right; otherwise, it must be crossed from right to left.

iii) Upon exiting the first bridge, the agent continues moving clockwise on the inner ring until it enters the bridge corresponding to its second variable, and then it continues in the same way with the bridge corresponding to its third variable.

iv) Upon exiting the third bridge, the agent must (i) perform a complete loop of the entire inner ring, moving in the clockwise direction, (ii) pass to the outer ring through the cell held by Λ_ν in the figure, (iii) traverse clockwise the entire outer ring and eventually terminate at the cell held by Z.

The agent executing the process annotated as Z must move in the counter-clockwise sense, initially traversing the outer ring, then entering the inner ring from the cell next to Λ_ν, and finally traversing this entire ring before terminating in its entering cell.

d) In state s, all the agents are on the first stages of the above described routes.

Clearly, the above construction can be polynomial with respect to the number of literals and clauses of the underlying 3-SAT problem. Next we establish that the considered 3-SAT problem instance has a solution *iff* the FREE-RANGE-RAS state s, that was defined through the above construction, is safe. For this, the reader should notice the following:

i) Since, in state s, process Z occupies a cell required by each process Λ_q for its completion, no Λ_q can complete until Z advances to another stage. Furthermore, since process Z moves on the two rings in a direction opposite to that of the motion of processes Λ_q, no state such that process Z is in one of the two rings while any other process Λ_q executes the last part of its route, as specified by item (c-iv) above, is safe.

ii) A little more reflection will reveal that the target state, where all processes have run to completion, is reachable from the considered state s, *iff* it is possible to reach a state s' such that all processes Λ_q are accomodated on the bridges and each bridge is occupied by processes that traverse it in the same direction.

Now notice that, by observation (ii), state s is safe *iff* there exists a safe state s' as characterized in (ii). Next we show that existence of such a safe state s' implies the existence of a satisficing literal subset K for the 3-SAT problem. Set K consists of all the literals X_i (resp., $\bar{X}_i$) which correspond to bridges that are non-empty of process instances in s' and are traversed from left to right (resp., from right to left). Indeed, set K satisfies property (1) posed by the 3-SAT problem, by means of observation (ii) above and the assumed safety of s'. It also satisfies property (2) posed by the 3-SAT problem, due to the specification of state s' and of K itself. It is easy to see that the reverse is also true, i.e., the existence of a satisficing set K for the 3-SAT problem enables the construction of the safe set s' postulated by observation (ii). Hence, it can be concluded that state s of the constructed RAS is safe *iff* the considered instance of 3-SAT has a solution.

V. Conclusions

In this paper we established the NP-completeness of the "state safety" problem arising in the context of some free-range multi-vehicle traffic systems that are encountered in modern technological applications. This result provides a formal theoretical base to ongoing efforts that seek to address the safe operation of these systems through suboptimal (i.e., non-maximally permissive) supervisory control policies that manage the underlying resource allocation. On the other hand, it is interesting to see how these results extend to other multi-vehicle traffic systems where different tessellation schemes may apply and/or the vehicle motion is confined to occur on more restrictive guidepath networks (e.g., like in the case of industrial AGV or monorail systems [16]). The systematic study of these extensions is part of our current investigations.

References

[1] S. M. La Valle and S. A. Hutchinson, "Optimal motion planning for multiple robots having independent goals," *IEEE Trans. on Robotics & Automation*, vol. 14, pp. 912–925, 1998.

[2] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: a study in multiagent hybrid systems," *IEEE Trans. on Automatic Control*, vol. 43, pp. 509–521, 1998.

[3] J. Lygeros, D. N. Godbole, and S. Sastry, "Verified hybrid controllers for automated vehicles," *IEEE Trans. on Automatic Control*, vol. 43, pp. 522–539, 1998.

[4] A. Bicchi and L. Pallottino, "On optimal cooperative conflict resolution of air traffic management systems," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, pp. 221–232, 2000.

[5] G. Inalhan, D. M. Stipanovic, and C. J. Tomlin, "Decentralized optimization, with application to multiple aircraft coordination," in *Proc. of CDC'02*. IEEE, 2002, pp. 1147–1155.

[6] S. Akella and S. Hutchinson, "Coordinating the motions of multiple robots with specified trajectories," in *Proc. of the 2002 IEEE Intl. Conference on Robotics & Automation*. IEEE, 2002, pp. 624–631.

[7] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos, "A feedback stabilization and collision avoidance scheme for multiple independent non-poin agents," *Automatica*, vol. 42, pp. 229–243, 2006.

[8] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized cooperative policy for conflict resolution in multivehicle systems," *IEEE Trans. on Robotics*, vol. 23, pp. 1170–1183, 2007.

[9] C. Belta, V. Isler, and G. Pappas, "Discrete abstractions for robot motion planning and control in polygonal environments," *IEEE Trans. on Robotics*, vol. 21, pp. 864–874, 2005.

[10] D. C. Conner, H. Choset, and A. A. Rizzi, "Flow-through policies for hybrid controller synthesis applied to fully actuated systems," *IEEE Trans. on Robotics*, vol. 25, pp. 136–146, 2009.

[11] E. Roszkowska, "Provably correct closed-loop control for multiple mobile robot systems," in *Proceedings of ICRA'05*. IEEE, 2005, pp. 2810–2815.

[12] S. A. Reveliotis and E. Roszkowska, "Conflict resolution in multi-vehicle systems: A resource allocation paradigm," in *Proceedings of IEEE CASE 2008*. IEEE, 2008, pp. –.

[13] S. A. Reveliotis, *Real-time Management of Resource Allocation Systems: A Discrete Event Systems Approach*. NY, NY: Springer, 2005.

[14] A. Kobetski, "Optimal coordination of flexible manufacturing systems with automatic generation of collision and deadlock-fee working schedules," Ph.D. dissertation, Chalmers University of Technology, Goteborg, Sweden, 2008.

[15] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*. Reading, MA: Addison-Wesley, 1979.

[16] M. P. Groover, *Fundamentals of Modern Manufacturing: Materials, Processes and Systems*. Englewood Cliffs, N.J.: Prentice Hall, 1996.